Uma comparação entre algoritmos baseados no modelo espaço-vetorial e algoritmos baseados em redes para o agrupamento de texto

João Vitor Sawada, Ivone Penque Matsuno, Rafael Geraldeli Rossi

[1]Universidade Federal de Mato Grosso do Sul (UFMS)
Caixa Postal 210, CEP 79620-080 – Três Lagoas – MS – Brazil

{joao.victor.sawada,ivone.matsuno,rafael.g.rossi}@ufms.br

Resumo. *A análise e extração de conhecimento contidos em grandes volumes de textos pode ser realizada através do agrupamento de textos, que separa uma coleção de textos em grupos nos quais os textos dentro de um mesmo grupo tendem a tratar do mesmo tema.. Entretanto, a maioria das abordagens de agrupamento considera que os textos são representados através de vetores de características tais como k-Means e Latent Dirichlet Allocation. As representações baseadas em redes, entretanto, tem demonstrado sucesso em tarefas envolvendo análise de texto. Em vista disso, o objetivo deste trabalho é comparar algoritmos baseados no modelo espaço-vetorial com algoritmos baseados em redes para a tarefa de agrupamento de texto. Foram considerado os algoritmos k-Means e DBSCAN como algoritmos baseados no modelo espaço-vetorial e os algoritmos Label Propagation, Greedy Modularity, e Girvan Newman como algoritmos baseados em redes. Os métodos foram comparados utilizando 21 coleções de textos de diferentes domínios. As coleções foram representadas no formato bag-of-words com o esquema de peso term frequency - inverse document frequency e as representações em redes foram geradas a partir da abordagem k-Nearest Neighbors. As métricas de avaliação utilizadas são: Acurácia, Pureza, Micro-Precisão, Micro-Revocação, Micro-F_1, Macro-Precisão, Macro-Revocação e Macro-F_1. Verificou-se que algoritmo de texto (DBSCAN) obteve os melhores resultados para a maioria das coleções de texto para a maior parte das métricas de avaliação.*

1. Introdução

Hoje em dia, uma enorme quantidade de dados circulam diariamente no ambiente digital [Turner et al. 2014]. Entre esses dados, alguns estão no formato textual, tais como artigos, notícias, *e-mails* e relatórios, sendo frequentemente usado para o armazenamento e disseminação de informações. As informações contidas nos textos podem ser úteis para compreender o comportamento humano, analisar a opinião pública, organizar informações e extrair conhecimento nos meios acadêmicos e empresariais [Biemann and Mehler 2014]. Entretanto, devido ao grande volume de textos produzidos e publicados atualmente, é humanamente impossível organizar, analisar e extrair conhecimentos embutidos manualmente. Como consequência, técnicas para automatizar tais tarefas com menor intervenção humana foram ganhando destaque nestes últimos anos [Aggarwal 2018].

Um dos meios mais viáveis de efetuar as tarefas mencionadas acima sem intervenção humana consiste no uso de algoritmos aprendizado de máquina não supervisionado para o agrupamento de textos. Esses algoritmos têm como objetivo extrair grupo de textos relacionados a um mesmo tema, tópico ou assunto, dispensando a necessidade do usuário informar as classes ou grupos os quais os documentos devem pertencer [Charu and Chandan 2013]. Os algoritmos de agrupamento, além de permitir a organização de coleções textuais, podem ser aplicados em diferentes outras áreas da mineração de textos, tais como a organização dos resultados retornados por um motor de busca [Zamir et al. 1997], geração de taxonomia de documentos da web [Koller and Sahami 1997], análise de dados espaciais [Bhadane and Shah 2020] e reconhecimento da emoção da fala [Kanwal and Asghar 2021].

Para que seja possível executar um algoritmo de agrupamento de dados, as coleções de textos devem possuir um formato estruturado. Normalmente, as coleções de textos são representadas utilizando o modelo espaço-vetorial em algoritmos como o k-*Means* e o *Latent Dirichlet allocation*. Neste modelo, os documentos são representados por vetores [Aggarwal 2018] e as dimensões correspondem aos termos ou atributos da coleção de textos [Rossi 2015]. Mesmo sendo muito utilizado, o modelo espaço-vetorial é pouco eficiente na representação de alguns aspectos dos textos e que podem ser úteis para a melhoria da performance dos algoritmos de aprendizado de máquina, como representar relações entre diferentes tipos de objetos e utilizar cadeias de relações [Zhou et al. 2004, Rossi et al. 2015]. Além disso, as representações baseados no modelo espaço-vetorial possuem alta dimensionalidade, o que torna o custo computacional alto, diminuindo a performance do agrupamento [Rossi 2015].

Por outro lado, como alternativa às representações baseadas no modelo espaço-vetorial, as representações em redes têm ganhado destaque nas últimas décadas [Tao et al. 2021, Zhang and Zhang 2020, Sun and Han 2012]. Com elas, é possível a representação de diversos tipos de relações entre os diferentes tipos de entidades dos textos, como documentos e termos [Rossi 2015]. Esses tipos de relações em uma rede possibilitam a captura de diferentes características nas coleções de textos e essas novas características permitem capturar padrões dificilmente capturados no modelo espaço-vetorial, o que pode proporcionar uma melhoria na qualidade do agrupamento [Ji et al. 2010].

Dados os benefícios dos algoritmos baseados em redes, ainda há uma lacuna quando se trata de uma comparação experimental entre algoritmos baseados em redes e algoritmos baseados no modelo espaço-vetorial para a tarefa de agrupamento de texto. Além disso, não há trabalhos na literatura que considerem os algoritmos apresentados neste trabalho em coleções textuais de diferentes domínios. Portanto, o objetivo deste trabalho é realizar uma extensa avaliação empírica comparando algoritmos de ambos os tipos. Portanto, para alcançar esse objetivo, 21 coleções de textos de diferentes domínios foram submetidas ao processamento dos seguintes algoritmos de detecção de comunidade: *Label Propagation* [Cordasco and Gargano 2011], *Greedy Modularity* [Newman 2004], *Girvan Newman* [Girvan and Newman 2002], *Edge Betweenness* [Brandes 2008], e algoritmos baseados no modelo espaço-vetorial: k-Means and DBSCAN [Tan et al. 2013]. Além disso, 8 métricas foram utilizadas para avaliar a qualidade dos grupos obtidos pelos algoritmos de detecção da comunidade. A partir dos resultados demonstrados neste trabalho, foi possível observar que o algoritmo DBSCAN,

um algoritmo baseado no modelo vetorial espacial, obteve o melhor desempenho em agrupamento de texto a partir da maioria das métricas para a maioria das coleções de texto.

O restante deste documento está estruturado da seguinte forma. A seção 2 apresentará o embasamento teórico e os trabalhos relacionadas encontrados na literatura. Na Seção 3, são apresentados detalhes do método de pesquisa utilizado neste trabalho. Na Seção 4, são apresentados os resultados e discussões. Finalmente, na Seção 5, apresentamos as conclusões e possibilidades para trabalhos futuros.

2. Embasamento Teórico e Trabalhos Relacionados

Essa seção apresenta o embasamento teórico e os trabalhos relacionados utilizados para o desenvolvimento deste trabalho.

2.1. Embasamento Téorico

Esta seção apresenta o embasamento teórico utilizado neste trabalho, são descritos os métodos de agrupamento para o modelo espaço-vetorial e para o modelo baseado em redes.

2.1.1. Representações Estruturadas

Esta seção apresenta os dois tipos de representações considerados neste trabalho: modelo de espaço-vetorial e representações baseadas em redes.

Representações baseadas no modelo espaço-vetorial Em uma representação baseada em modelo de espaço-vetorial, os documentos ou instâncias do conjunto de dados são representados por um vetor, e cada dimensão está relacionada a um atributo do texto. Uma das representações mais tradicionais baseadas no espaço vectorial é a *bag-of-words* [Aggarwal 2018]. Nessa representação, é feito o uso de palavras simples como termos, gerando uma matriz documento-termo, essa matriz apresenta alta dimensionalidade, visto que nas coleções de textos existe um grande número de palavras e alta esparsidade, isto ocorre em razão do fato de que a das palavras ocorrem apenas em uma pequena parte dos documentos [Rossi 2015]. É possível visualizar essa representação na Tabela 1, no qual o conjunto de documentos é representado por $\mathcal{D} = \{d_1, d_2, ..., d_m\}$, o conjunto de termos por $\mathcal{T} = \{t_1, t_2, ..., t_n\}$, e w_{d_i,t_j} representa o peso de um termo t_j em um documento d_i. Neste trabalho, foi utilizado o esquema de peso *term frequency - inverse document frequency*, pelo qual os pesos das relações são definidos pela frequência do termo e pelo inverso da frequência do documento.

Para reduzir o número de termos e melhorar os resultados das técnicas de mineração de texto, é feito o pré-processamento das coleções de textos. Dentre algumas das técnicas de pré-processamento estão: a padronização de caixas, a simplificação de palavras e a remoção de *stopwords*, que são palavras consideradas irrelevantes, tais como pronomes e artigos, quando se busca de padrões em algoritmos de aprendizado de máquina [Aggarwal 2018].

Tabela 1. Representação no espaço-vetorial para *m* documentos e *n* atributos.

	t_1	t_2	t_3	$\cdots$	t_n
d_1	w_{d_1,t_1}	w_{d_1,t_2}	w_{d_1,t_3}	$\cdots$	w_{d_1,t_n}
d_2	w_{d_2,t_1}	w_{d_2,t_2}	w_{d_2,t_3}	$\cdots$	w_{d_2,t_n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
d_m	w_{d_m,t_1}	w_{d_m,t_2}	w_{d_m,t_3}	$\cdots$	w_{d_m,t_n}

Representação baseada em redes As coleções de textos também podem ser representadas através de redes [Rossi 2015]. As representações baseada em redes permite a integração de vários aspectos, tais como topologia, estatística e gramática dentro de um único modelo e em um formalismo tratável matematicamente [Blanco and Lioma 2012].

Uma rede pode ser formalmente definida através do conjunto $\mathcal{N} = \langle \mathcal{O}, \mathcal{R}, \mathcal{W} \rangle$, onde $\mathcal{O}, \mathcal{R}, \mathcal{W}$ correspondem respectivamente aos conjuntos de objetos, relações e peso nas relações [Rossi et al. 2013]. Todas as relações de uma rede podem ter o mesmo peso, isto é, $o_i, oj \in \mathcal{O}, w_{o_i,o_j} = 1$ se $\exists r_{o_i,o_j} \in \mathcal{R}$. Esses tipos de redes são denominadas de redes não ponderadas. Em contrapartida, redes que consideram os pesos das relações em suas arestas, isto é, $\exists r_{o_i,o_j}$, podem ser usados números reais para representar o grau da relação entre os objetos. Essas redes são conhecidas como redes ponderadas.

As representações baseadas em redes permite extrair padrões de classe que dificilmente são capturados no modelo espaço-vetorial [Breve et al. 2012].Existem diferentes tipos de redes que podem ser usadas para representar qualquer coleção de textos, tais como redes bipartidas (objetos correspondem a documentos e termos, onde os termos estão conectados aos documentos nos quais eles ocorrem), redes de termos (objetos são os termos e relações entre termos podem ser sintáticos, semânticos, por ordem de ocorrência ou similaridade), ou uma rede de documentos (objetos que representam os documentos e relações podem ser dados explicitamente, tais como hyperlinks ou citações, ou implicitamente, tais medidas de similaridade) [Rossi et al. 2015].

Uma forma típica de representar coleções de textos é através da rede de documentos [Rossi 2015]. Nesta rede, os documentos são representados por $\mathcal{O} = \mathcal{D}$. As relações das redes podem ser representadas de forma explícita, como na forma de hiperlinks e citações [Getoor 2005]. Entretanto, estudos adicionais mostraram que relações implícitas, como redes de semelhanças, oferecem melhores resultados [Angelova and Weikum 2006]. Portanto, as redes de similaridade foram escolhidas para representar as relações entre documentos neste trabalho, e particularmente a semelhança de cossenos como medida de similaridade. Na Figura 1 é ilustrado um exemplo de rede de documentos baseada em similaridade [Gualdi and Rossi 2019], no qual cada retângulo representa o documento de uma coleção e cada aresta indica uma relação entre os pares de documentos dado por algum método de construção de rede. A $\text{Sim}(\text{Doc } i, \text{Doc } j))$ denota o peso da aresta, o qual é dado pela similaridade entre os documentos i e j.

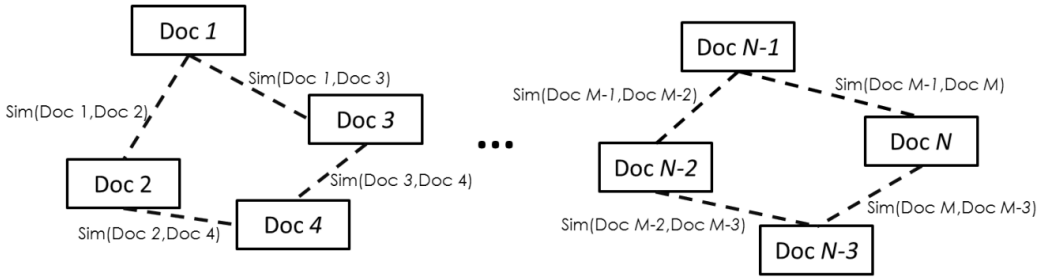


Figura 1. Ilustração de uma rede de documentos baseada em similaridade.

2.1.2. Algoritmos de agrupamento

Esta seção apresenta dois tipos de algoritmos de agrupamento cobertos neste trabalho: os algoritmos baseados no modelo espaço-vetorial e os algoritmos baseados em redes. Neste trabalho, tratamos os termos *cluster* e comunidade indiscriminadamente.

Algoritmos baseados no modelo espaço-vetorial Os algoritmos baseados no modelo espaço-vetorial utilizados neste trabalho estão divididos em duas categorias: algoritmos com abordagem baseada em protótipos e algoritmos com abordagem baseada em densidade [Tan et al. 2013].

Algoritmos com abordagem baseada em protótipos Os algoritmos baseados em protótipos agrupam os dados com base na atribuição de cada observação a seu protótipo mais próximo (*centroid*, *medoid*, etc.) [Zararsiz 2014]. O algoritmo baseado em protótipos utilizado neste trabalho é o k-*Means*. Neste algoritmo, o parâmetro k é o número de *clusters* ou grupos desejados. A partir da definição de k, seu objetivo é gerar k grupos posicionando k centroides, que são inicializados em posições aleatórias. Cada centroide representa o ponto médio do grupo, no qual os documentos mais próximos compõem os *clusters*. A cada iteração do algoritmo, os documentos são atribuídos ao centroide mais próximo, e então a posição do centroide é recalculada. Este ciclo é repetido até iterações de x ou até um critério de parada, por exemplo, até que não haja alterações nos grupos em uma determinada iteração.

Cada iteração do k-Means procura reduzir a coesão, ou seja, a proximidade dos documentos do centroide com o *cluster* deandrade:2017. Dado um conjunto de *clusters* $\mathcal{C} = C_1, C_2, ..., C_k$, a coesão de um *cluster* C_i pode ser representada pela Equação 1:

$$\text{coesão} = \sum_{C_i \in C} \sum_{d_j \in C_i} \cos(w_{d_j}, c_i) \tag{1}$$

no qual c_i representa o centroide de um conjunto de documentos e w_{d_j} representa o peso de um documento de d_i.

Algoritmos com abordagem baseada em densidade Os algoritmos baseados em densidade agrupam os dados com base em sua densidade, segmentando regiões de alta ou baixa densidade. Em uma abordagem centralizada, sua densidade é estimada por um determinado ponto no conjunto de dados por meio da contagem do número de pontos de dados dentro de um raio determinado [Tan et al. 2013]. Neste método, a densidade de qualquer ponto dependerá do raio determinado, ou seja, se o raio for suficientemente grande, cada ponto terá uma densidade de m, que é o número de pontos no conjunto de dados. Da mesma forma, se o raio for muito pequeno, então todos os pontos terão uma densidade de 1. Além disso, a abordagem baseada no centro classifica um ponto com base em sua região, que pode ser um ponto central, um ponto de fronteira ou um ponto de ruído.

Um ponto central é um ponto dentro de uma região densa, ou seja, uma região altamente povoada. Um ponto é considerado um ponto central quando a distância do ponto até a borda de sua vizinhança (*Eps*) e o número mínimo de vizinhos (*MinPts*) são delimitados por um limite. Um ponto de fronteira é um ponto que não é um ponto central, mas que está dentro da vizinhança de um ou mais pontos centrais. E por último, um ponto de ruído é qualquer ponto que não seja um ponto central nem um ponto de fronteira, ou seja, um ponto que não esteja na vizinhança de nenhum dos pontos centrais.

O algoritmo baseado na densidade utilizado neste trabalho é o *DBSCAN*, um algoritmo com uma abordagem centralizada que encontra amostras de núcleo de alta densidade e expande os *clusters* baseados nelas. Se dois pontos estão próximos um do outro, e esta proximidade é definida pela distância *Eps* de um ponto ao outro, eles pertencem ao mesmo aglomerado. Da mesma forma, se um ponto de fronteira está suficientemente próximo de outro ponto central, ele pertence ao mesmo *cluster* que o ponto central. Além disso, todos os pontos de ruído são descartados e não farão parte de nenhum *cluster*.

Algoritmos baseados em redes Os algoritmos baseados em redes utilizados neste trabalho estão divididos em duas categorias em três categorias: algoritmos de propagação de rótulos, algoritmos de modularidade e algoritmos de centralidade.

Algoritmos baseados em propagação de rótulos Recentemente, o algoritmo *Label Propagation* tem mostrado bons resultados no agrupamento de textos [Gualdi and Rossi 2019]. Este algoritmo atribui inicialmente um rótulo único a todos os nós da rede. Em seguida, é realizado um processo iterativo para que os conjuntos de nós conectados sejam capazes de chegar a um acordo sobre algum rótulo que dê origem à comunidade. No final de cada etapa do processo, cada nó atualiza sua etiqueta para uma nova, correspondente à etiqueta mais frequente de seus vizinhos.

Na definição formal, cada objeto pode ter um rótulo associado presente no conjunto, $\mathcal{L} = \{l_1, l_2, ..., l_{|\mathcal{O}|}\}$. O rótulo de um objeto o_i é dado por $label(o_i)$, que é calculado através da seguinte fórmula:

$$\textit{label}(o_i) = \arg\max_{l_k \in \mathcal{L}} \sum_{o_i, o_j \in \mathcal{O}} w_{o_i,o_j} \delta(o_j, l_k), \tag{2}$$

no qual $\delta(o_j, l_k)$ devolve 1 se o objeto o_j tiver um rótulo de l_k, e 0 de outra forma. O algoritmo é executado iterativamente até que os rótulos permaneçam inalterados. No final do processo, objetos com o mesmo rótulo pertencerão ao mesmo grupo. No caso de um empate, ou seja, existem múltiplos rótulos com a mesma frequência, os rótulos são escolhidos aleatoriamente.

O processo de atualização das etiquetas pode ser síncrono ou assíncrono [Raghavan et al. 2007]. No modelo síncrono, cada nó computa seu próprio rótulo considerando o rótulo de seus vizinhos de uma iteração anterior. Como não há dependências entre os rótulos pertencentes à mesma etapa, cada etapa de propagação de rótulos pode ser realizada em paralelo nos vértices [Cordasco and Gargano 2011]. Entretanto, foi demonstrado que a atualização síncrona pode resultar em uma oscilação cíclica das etiquetas [Raghavan et al. 2007]. Isto ocorre principalmente quando a rede é bipartida ou em forma de estrela, mas pode ocorrer em outras situações.

Para evitar possíveis ciclos e garantir a finalização do algoritmo, é possível usar a abordagem assíncrona na propagação de rótulos, na qual cada nó é atualizado usando o rótulo da iteração atual. Embora esta abordagem reduza a chance de oscilações cíclicas, ela requer que cada etapa do algoritmo seja sequencial. Além disso, a abordagem assíncrona é instável, pois cada sequência de atualização de rótulos é escolhida aleatoriamente e pode gerar resultados diferentes com as mesmas configurações iniciais.

Algoritmos baseados em modularidade A modularidade em uma rede é uma medida para calcular a força das comunidades. Dada uma rede, onde cada um de seus nós pertence a uma comunidade, a modularidade da rede é definida por:

$$Q(C) = \frac{1}{2|\mathcal{R}|} \sum_{o_i, o_j \in \mathcal{O}} \left(A_{o_i,o_j} - \frac{\textit{degree}(o_i)\textit{degree}(o_j)}{2|\mathcal{R}|} \right) \delta_{\textit{cluster}(o_i),\textit{cluster}(o_j)} \qquad (3)$$

em que $\mathcal{A}$ representa a matriz de adjacência da rede, *degree*(o_i) e *degree*(do_j) denotam o grau dos nós o_i e o_j, tal que o grau é a soma dos pesos de borda adjacentes ao nó. Além disso, o $\delta_{\textit{cluster}(o_i),\textit{cluster}(o_j)}$ representa a função de *Kronecker delta*, que limita a soma aos pares de vértices da mesma comunidade, tendo valor 1 se o_i ou o_j se seus argumentos forem iguais e 0 caso contrário [Ganji et al. 2015].

Entretanto, a maximização da modularidade através da força bruta é computacionalmente inviável [Newman 2004]. Em razão disto, a heurística é empregada para maximizar a modularidade de uma forma não otimizada. Um exemplo de maximização da modularidade baseada na heurística é o algoritmo *Greedy Modularity*, projetado para aumentar a modularidade da rede de uma forma gananciosa [Newman 2004]. Neste algoritmo, cada nó é inicializado em uma comunidade diferente, e a cada iteração, um par de nós que mais aumenta a modularidade é atribuído à mesma comunidade, e então toda a modularidade da rede é atualizada. Este processo é repetido até que cada par seja indivisível, e nenhuma outra melhoria na modularidade é possível. No final, o conjunto de comunidades que apresentou o maior valor de modularidade durante as iterações do algoritmo é considerado a solução.

Algoritmos baseados em centralidade As medidas de centralidade fazem suposições implícitas sobre como o tráfego flui em uma rede. Estes indicadores de centralidade apontam para os principais nós ou bordas de uma rede. Neste documento, foi utilizado um algoritmo de detecção de comunidades baseado na intermediação entre as arestas [Freeman 1977]. Esta medida de centralidade é baseada nos caminhos mais curtos em um grafo, de modo que seu objetivo é encontrar a aresta mais central da rede, ou seja, as arestas que pertencem ao maior número de caminhos mínimos entre os pares de nós da rede. A interseção de uma borda pode ser formalmente definida a partir de:

$$C(\mathcal{R}_i) = \sum_{s \neq v \neq t \in \mathcal{R}} \frac{\sigma_{\text{st}(\mathcal{R}_i}\big)}{\sigma_{\text{st}}}, \tag{4}$$

em que o_s e o_t são respectivamente o ponto de partida e o ponto final que representam um caminho na rede. Se o caminho mais curto é denotado por $\sigma_{st} = \sigma_{ts}$, de tal forma que $s \in \mathcal{R}$ e $t \in \mathcal{R}$, enquanto $\sigma_{st}(\mathcal{R}_i)$ denota o número de caminhos mais curtos de s a t que passam por $\mathcal{R}_i$.

O algoritmo Girvan Newman é um algoritmo de detecção de comunidades que remove as bordas de uma rede de forma progressiva [Girvan and Newman 2002]. Normalmente, a borda a ser removida é a borda com maior intermediação. Se a rede tem poucos grupos conectados (com poucas arestas conectando o grupo), então o caminho mais curto deve passar por este pequeno número de arestas. Consequentemente, estes caminhos terão um alto valor de intermediação, sendo removidos e gerando as comunidades na rede.

Neste trabalho, será considerada uma variante normalizada do algoritmo *Girvan Newman* [Brandes 2008], através do qual receberá os valores de entrelaçamento já normalizados. A normalização pode ser definida por:

$$\frac{2}{(\mathcal{O} \cdot (\mathcal{O} - 1))}, \tag{5}$$

em que $\mathcal{O}$ é o número de nós na rede. Ao contrário do algoritmo *Girvan Newman*, esta variante considera os pesos das relações dos nós. Entretanto, estas relações devem ter pesos maiores que zero, pois se forem iguais a zero, uma infinidade de caminhos com o mesmo comprimento será produzida entre os pares de nós. Além disso, nesta variante, os valores de centralidade são padronizados pela divisão da medida de centralidade da borda mais central. Além disso, um ruído é adicionado à centralidade, que é um valor aleatório entre 0 a 1, mudando assim o ranqueamento das bordas e gerando resultados possivelmente diferentes dos do algoritmo *Girvan Newman*.

2.2. Trabalhos relacionados

Há vários artigos na literatura que comparam o desempenho dos algoritmos de agrupamento, utilizando conjuntos de dados naturais [Massey 2005, Jalil et al. 2016] e conjuntos de dados artificiais [Rodriguez et al. 2019]. Esta seção é dedicada às características desses estudos.

[Rodriguez et al. 2019] realizou uma comparação sistemática de 11 métodos de agrupamento diferentes: k-Means, *CLARA*, *single linkage*, *complete linkage*, *Ward's*

method, *weighted average linkage*, *EM*, *HCMODEL*, *spectral*, *subspace*, *OPTICS* e *DBSCAN*, utilizando o *Adjusted Rand index* como métrica de avaliação. O conjunto de dados utilizado foi um conjunto de dados artificial gerado aleatoriamente. Descobriu-se que os resultados do algoritmo de agrupamento *spectral* mostraram o melhor desempenho quando usado com parâmetros padrão. No entanto, os algoritmos *EM*, algoritmos hierárquicos, k-Means e *subspace* alcançaram um desempenho semelhante quando alguns dos parâmetros foram ajustados.

[Jalil et al. 2016] realizou um estudo comparativo focalizando o processamento de grandes quantidades de dados, usando 5 diferentes algoritmos de agrupamento: *k-Means*, *Global k-Means*, *Fast Global k-Means*, *Two Level k-Means* e *k-Means type subspace clustering algorithm (FW-k-Means)*, usando o *Davies-Bouldin index*, erro quadrático e tempo de execução como métricas de avaliação. A comparação entre estes algoritmos foi feita considerando dados textuais reais da web, através de feeds RSS. Verificou-se que o algoritmo *FW-k-Means* apresentou os melhores resultados quando comparado à maioria dos algoritmos testados, foi capaz de lidar com dados grandes e esparsos, porém, seu tempo de processamento é alto. O algoritmo de *k-Means* demonstrou rápida convergência para agrupamento de texto. E por fim, o algoritmo de dois níveis *k-Means* teve resultados semelhantes aos do algoritmo *k-Means*, porém, a escolha de seu valor limiar ainda é um problema, demonstrando assim sua ineficiência no processamento de grandes volumes de dados.

[Massey 2005] avaliou e comparou 3 algoritmos de agrupamento: *Adaptative Resonance Theory neural network* (*ART*), *k-Means* e *spherical k-Means*, usando o F_1 como métrica de avaliação. Os conjuntos de dados utilizados foram *Reuters-21578*, usando *Modified Apte (ModApte) Split* e o conjunto de dados *HD-49*, um subconjunto do conjunto de dados *Ohsumed*. Verificou-se que para o conjunto de dados *Reuters*, o algoritmo que mostrou os melhores resultados foi o algoritmo *ART*, enquanto para o conjunto de dados *HD-49*, o algoritmo com os melhores resultados foi o algoritmo de *k-Means*.

Analisando os trabalhos apresentados, é possível observar que o algoritmo de *k-Means* é comumente usado no agrupamento de textos, mas ainda há uma lacuna na literatura quando se trata de uma avaliação comparativa entre os algoritmos baseados no modelo espaço-vetorial e os algoritmos baseados em redes para o agrupamento de texto. Além disso, o trabalho envolvendo a comparação entre algoritmos de agrupamento realiza a comparação em um pequeno número de conjuntos de dados e com uma gama estreita de domínios.

3. Método de pesquisa

Nesta seção, apresentamos os detalhes do método de pesquisa adotado neste trabalho. O método de pesquisa diz respeito a 3 etapas: (i) coleta de coleções de textos; (ii) préprocessamento de coleções de textos, (iii) definição dos parâmetros e execução dos algoritmos de agrupamento, e (iv) avaliação dos resultados do agrupamento. Nas subseções seguintes, são apresentados os detalhes destas etapas.

3.1. Coleções de texto

Para as avaliações experimentais, foram utilizadas 21 coleções de textos de diferentes domínios [Rossi et al. 2013]. A Tabela 2 apresenta os detalhes das coleções textuais,

como o nome da coleção, número de documentos $|\mathcal{D}|$, número de termos $|\mathcal{T}|$, número médio de termos por documento $\overline{|\mathcal{T}|}$ e o domínio contido na coleção textual. As coleções são pré-processadas no formato *bag-of-words* utilizando o esquema de peso *term frequency - inverse document frequency*. Além disso, cada coleção de textos está associada a um dos seguintes domínios:

- Análise dos Sentimentos (AS)
- Artigos de Notícias (AN)
- Documentos Científicos (DC)
- Documentos Médicos (DM)
- Páginas Web (PW)
- Recuperação de Informações (RI)

Tabela 2. Tabela de coleções de textos utilizadas na avaliação experimental

Coleção	$\|\mathcal{D}\|$	$\|\mathcal{T}\|$	$\overline{\|\mathcal{T}\|}$	**Domínio**
CSTR	299	1726	54.27	DC
Dmoz-Health-500	18500	8303	54.27	PW
Hitech	2301	12942	141.93	AN
Irish-Sentiment	1660	8659	112.64	AS
La1s	3202	13196	144.64	AN
La2s	3075	12433	144.83	AN
Oh0	1003	3183	52.50	DM
Oh5	918	3013	54.43	DM
Oh10	1050	3239	55.63	DM
Oh15	913	3101	59.30	DM
Re0	1504	2887	51.72	AN
Re1	1657	3759	52.69	AN
Reviews	4069	22927	183.10	AN
SyskillWebert	334	4340	93.15	PW
Tr11	414	6430	281.66	RI
Tr12	313	5805	273.59	RI
Tr21	336	7903	469.86	RI
Tr23	204	5833	385.29	RI
Tr31	927	10129	268.49	RI
Tr45	690	8262	280.58	RI
WAP	1560	8461	141.33	PW

3.2. Pré-processamento

No pré-processamento das coleções de textos, palavras isoladas foram consideradas como termos, *stopwords* foram removidas, os termos foram fixados usando o algoritmo de Porter [Porter 1980], *tags HTML* foram removidas, e somente termos com frequência de documentos ≥ 2 foram considerados. Para ponderar os termos em documentos, foram utilizados termos frequência e frequência inversa de documentos.

Para evitar a geração de redes altamente desconectadas e consequentemente aplicar os algoritmos baseados em redes de forma justa, os resultados serão gerados apenas para redes que satisfaçam a condição $|\mathcal{G}| \leq \sqrt{|\mathcal{D}|}$, onde $\mathcal{G}$, e $\mathcal{D}$ correspondem respectivamente aos grupos gerados pelos algoritmos de detecção da comunidade e ao conjunto de documentos.

3.3. Algoritmos e parâmetros

Os algoritmos apresentados foram avaliados experimentalmente usando a biblioteca *NetworkX* [Hagberg et al. 2008] para os algoritmos baseados em redes e a biblioteca *scikit-learn* [Pedregosa et al. 2011] para os algoritmos baseados no modelo espaço-vetorial. Para todos os algoritmos de agrupamento baseados em redes, foi utilizada a rede k*-vizinhos mais próximos* (K*-NN*) a semelhança de cossenos como medidas de distância. Além disso, diferentes redes k*-NN* foram geradas com o k número de vizinhos variando de 3 até 25, no qual $(\forall k = 2 \cdot z + 1 | z \in [1...25])$. Os algoritmos de agrupamento, juntamente com seus respectivos parâmetros, são:

- ***Label Propagation*** **(LP)**: com atualizações de rótulos síncronas e assíncronas, com pesos nas relações somente na versão assíncrona.
- ***Greedy Modularity*** **(GM)**: com e sem pesos nas relações.
- ***Girvan Newman*** **(GN)**: usando a intermediação como medida de centralidade.
- ***Edge betweenness*** **(EB)**: com e sem peso nas relações, utilizando os valores de centralidade normalizados e padronizados e com adição de um rúido aleatório.
- k***-Means*** **(KM)**: $k \in [3...16]$, com os centroides inicializados em posições aleatórias.
- ***DBSCAN*** **(DB)**: *eps* $\in [0.25, 0.5, 0.75]$ e *min_samples* $= 5$, usando a distância de cossenos como medida de distância.

3.4. Avaliação experimental

Nas coleções de textos (Seção 3.1), cada documento está associado a um rótulo, correspondente a seu tema naquele domínio. Ou seja, dado um conjunto de rótulos $\mathcal{L}$, cada $\mathcal{L}_i$ está associado a um documento d_j. Estes rótulos podem ser usados para avaliar a qualidade dos grupos obtidos após o agrupamento, utilizando as seguintes métricas de desempenho implementadas pela biblioteca *scikit-learn* [Pedregosa et al. 2011]: Acurácia, Pureza, Micro-Precisão, Micro-Revocação, Micro-F_1, Macro-Precisão, Macro-Revocação e Macro-F_1 [Tan et al. 2013]. Foram executadas 10 iterações para cada configuração de algoritmo devido a fatores de aleatoriedade incorporados. No final, a média dos resultados obtidos considerando os resultados das 10 iterações.

4. Resultados e Discussões

Na Figura 2, são apresentados exemplos de agrupamento de documentos de todos os algoritmos de agrupamento para a coleção de textos *CSTR*. Para isso, cada documento da coleção de textos foi representado como um nó e cada rótulo associado a esse documento foi representado por uma cor. Para os algoritmos baseados em redes, foi utilizada a rede k-*NN* juntamente com a similaridade de cossenos como forma de representação dos documentos, para os algoritmos baseados no modelo espaço-vetorial foi utilizado a representação estruturada *bag-of-words*. Cada algoritmo gerou n grupos que foram utilizados para prever o valor de cada rótulo com base no rótulo mais frequente.

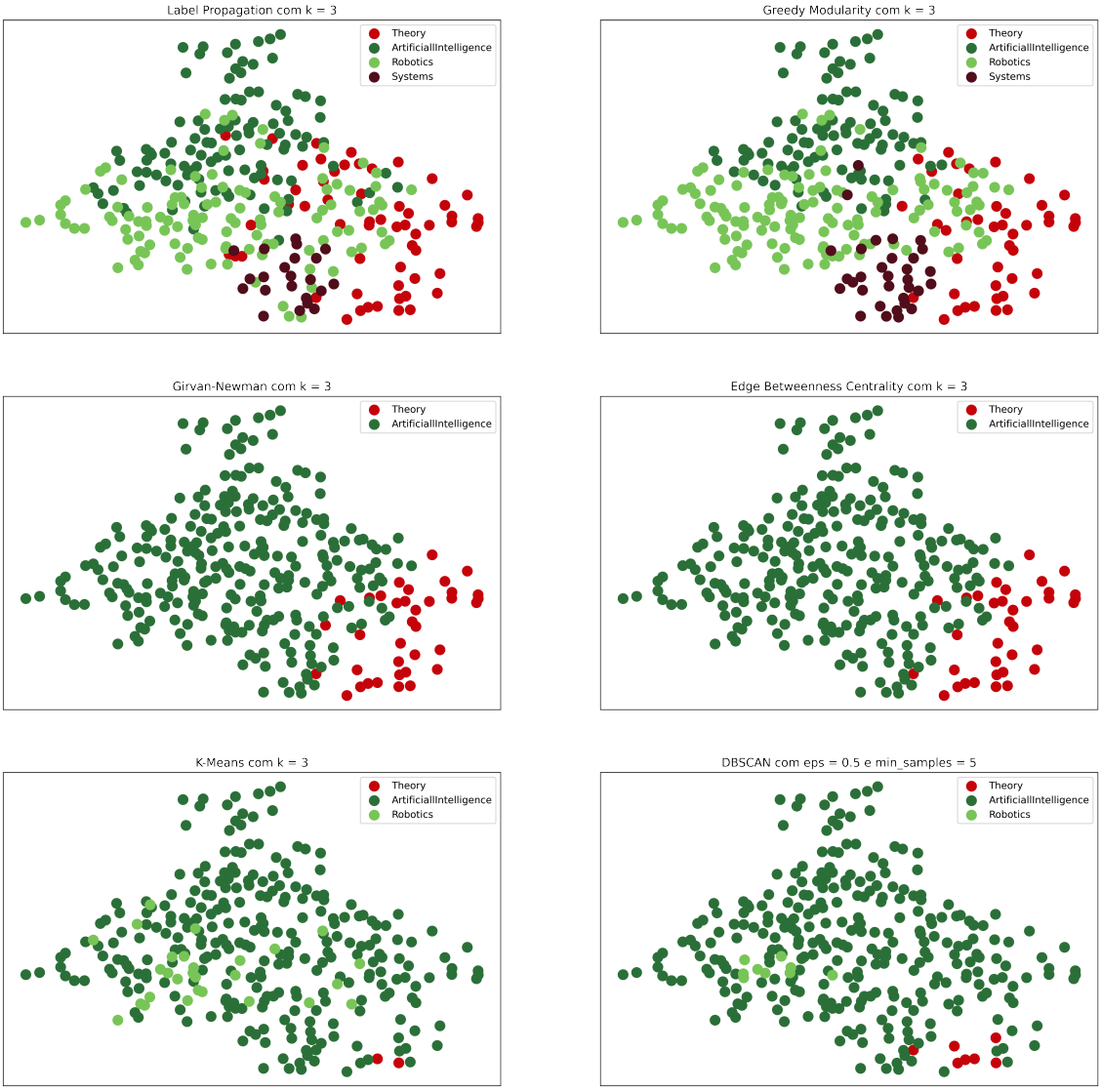


Figura 2. Agrupamento de cada algoritmo para a coleção de textos CSTR.

Analisando a figura, é possível observar que os algoritmos baseados em redes obtiveram grupos mais heterogêneos, capturando padrões que não foram capturados nos algoritmos baseados no modelo espaço-vetorial. Os algoritmos baseados em redes, *Label Propagation* e *Greedy Modularity* obtiveram um agrupamento semelhante, assim como os algoritmos baseados no modelo espaço-vetorial, k-*Means* e *DBSCAN*. Também é possível

observar que os algoritmos baseados no modelo espaço-vetorial rotularam a grande maioria dos documentos como pertencentes a classe *ArtificialIntelligence*, já os algoritmos baseados em redes rotularam os documentos de forma mais balanceada.

Na Tabela 3, são apresentados os resultados[1] indicados pela métrica Macro F_1 para os algoritmos: *Label Propagation*, *Greedy Modularity*, k-*Means* e *DBSCAN*, uma vez que estes algoritmos obtiveram resultados nas 21 coleções de texto. É possível observar que os algoritmos com os melhores resultados considerando a métrica Macro F_1 foi o algoritmo *DBSCAN* e o algoritmo k-*Means*, que empatados no número de vitórias, ambos com 7 vitórias. Também é possível notar que, embora não tenham conseguido alcançar os resultados dos algoritmos baseados no modelo espaço-vetorial, os algoritmos baseados em redes *Label Propagation* e *Greedy Modularity* apresentaram resultados semelhantes, diferindo apenas por uma vitória cada um.

Tabela 3. Maior resultado obtido a partir da métrica Macro-F_1 para todos os parâmetros de configurações dos seguintes algoritmos de detecção de comunidades: *Label Propagation*, *Greedy Modularity*, k-*Means* e *DBSCAN*.

Coleção	LP	GM	KM	DB
CSTR	**0.454**	0.433	0.304	0.110
Dmoz-Health-500	**0.119**	0.102	0.092	0.099
Hitech	0.177	**0.332**	0.235	0.248
Irish-Sentiment	0.175	**0.472**	0.384	0.149
Reviews	0.264	0.311	**0.338**	0.208
SyskillWebert	**0.860**	0.337	0.339	0.307
La1	0.178	**0.180**	0.175	0.170
La2	**0.376**	0.325	0.179	0.237
Oh0	0.089	0.104	**0.125**	0.070
Oh10	0.106	0.058	**0.155**	0.064
Oh15	0.087	0.090	**0.112**	0.064
Oh5	0.100	0.098	**0.117**	0.079
Re0	0.073	0.081	0.119	**0.160**
Re1	0.044	0.000	0.075	**0.222**
Tr11	0.100	0.086	0.146	**0.243**
Tr12	0.100	0.138	0.135	**0.237**
Tr21	0.162	0.178	**0.287**	0.195
Tr23	0.190	0.192	**0.238**	0.180
Tr31	0.146	0.171	0.214	**0.305**
Tr45	0.104	0.120	0.133	**0.370**
WAP	0.034	0.053	0.071	**0.223**
Média de Macro-F_1	0.181	0.184	0.189	0.188
Ranking	3	4	1	2
Número de vitórias	4	3	7	7

Na Tabela 4, são apresentados as coleções nas quais os resultados foram obtidos para todos os algoritmos considerados na avaliação experimental. Neste caso, o algoritmo que mostrou os melhores resultados foi o algoritmo *Greedy Modularity*, obtendo

[1]Todos os resultados apresentados neste trabalho, bem como a estrutura desenvolvida para obter os resultados, podem ser encontrados inteiramente em `https://nyvemm.github.io/results_clustering_algorithm_network/`.

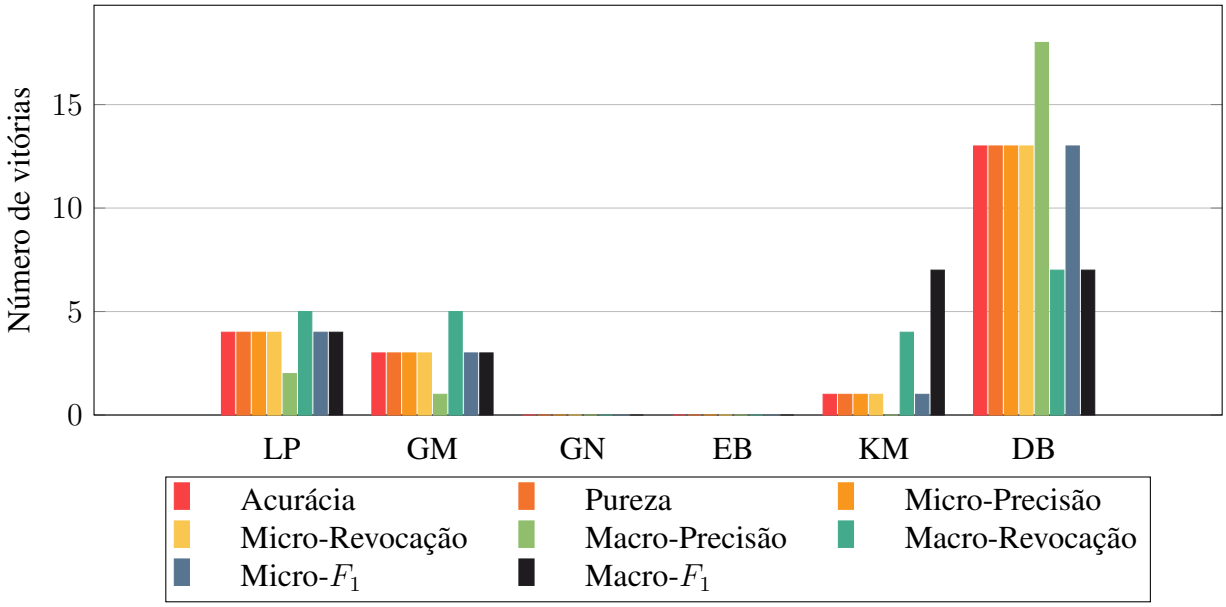
2 vitórias e tendo uma média da métrica Macro-F_1 maior que os outros algoritmos. É notável que, os algoritmos baseados em medidas de centralidade não obtiveram um bom desempenho, não conseguindo atingir nenhuma vitória. Ao fazer uma comparação entre os algoritmos baseados em medidas de centralidade, o algoritmo com os melhores resultados é o algoritmo *Edge Betweeness*.

Tabela 4. Maior resultado obtido a partir da métrica Macro-F_1 para coleções que possuem os resultados de algoritmos de centralidade.

Coleção	LP	GM	GN	EB	KM	DB
Hitech	0.177	**0.332**	0.059	0.059	0.235	0.248
Reviews	0.264	0.311	0.088	0.088	**0.338**	0.208
La1	0.178	**0.180**	0.097	0.175	0.175	0.170
Tr31	0.146	0.171	0.116	0.117	0.214	**0.305**
Média de Macro-F_1	0.208	0.284	0.090	0.119	0.240	0.233
Ranking	4	1	6	5	2	3
Número de vitórias	0	2	0	0	1	1

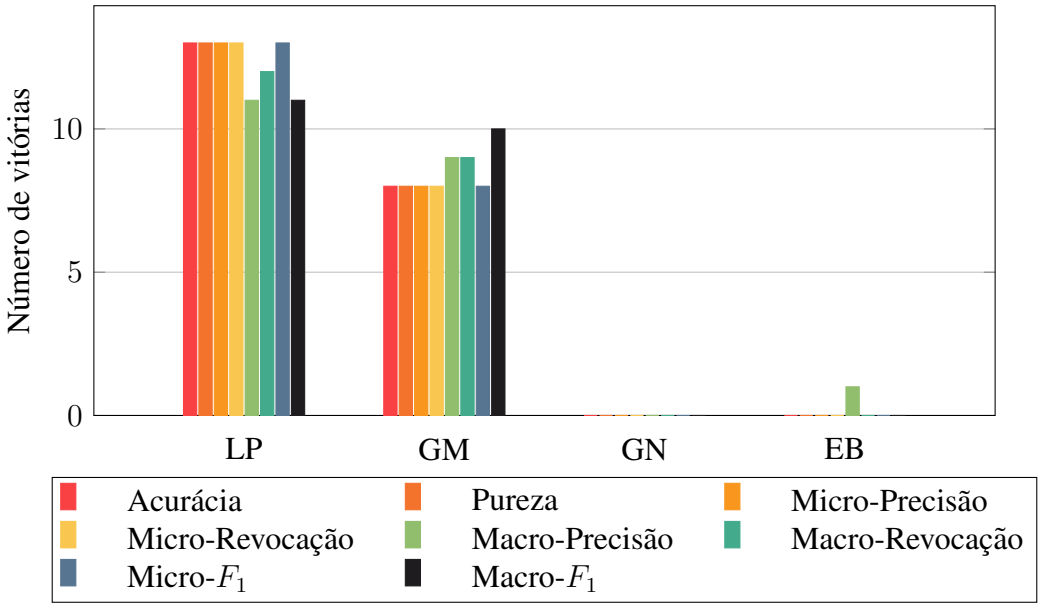
Nos próximos parágrafos, são apresentadas figuras que resumem os dados obtidos nos resultados. Na Figura 3, o número de vitórias de cada algoritmo de agrupamento é apresentado, considerando as diferentes métricas de avaliação utilizadas neste trabalho. Analisando a figura, é possível observar que o algoritmo com o maior número de vitórias é o algoritmo *DBSCAN*, que foi vitorioso em todas as métricas com exceção da métrica Macro-F_1, na qual ocorreu um empate com o algoritmo k-*Means*. Também é interessante observar que os algoritmos baseados em redes com exceção dos algoritmos baseados em medidas de centralidade tiveram um resultado melhor do que o algoritmo k-*Means* na maioria das métricas.

Figura 3. Número de vitórias de cada algoritmo de agrupamento para todas as métricas de avaliação.



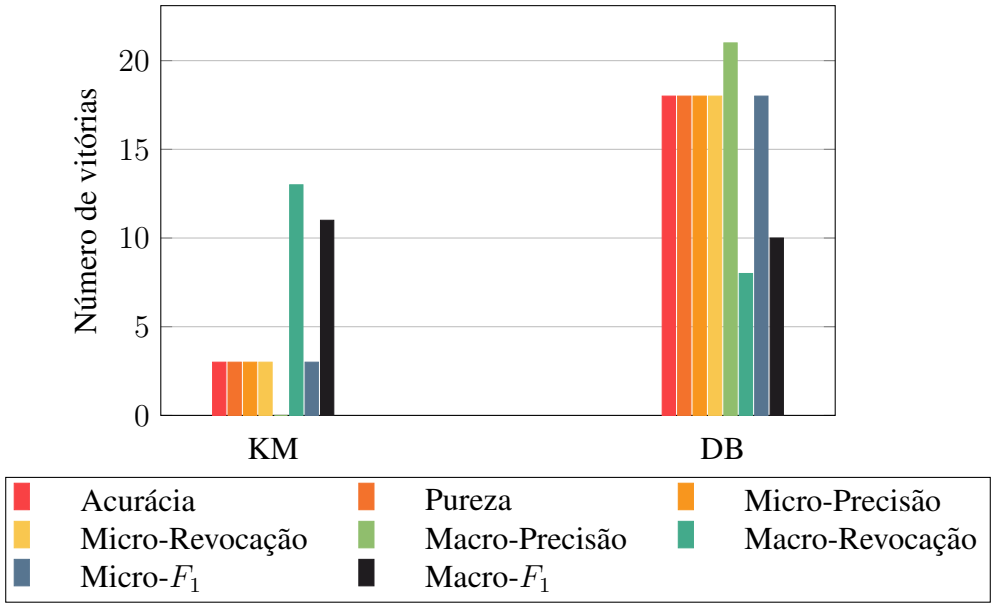
Na Figura 4, o número de vitórias de cada algoritmo baseado em redes é apresentado, considerando as diferentes medidas utilizadas neste trabalho. Com base em sua observação, Analisando a figura, é possível observar que o algoritmo com maior número de vitórias entre os algoritmos baseados em redes é o algoritmo *Label Propagation*. Embora não conseguindo superar o algoritmo *Label Propagation*, o algoritmo *Greedy Modularity* também alcançou uma boa taxa de vitórias e se manteve competitivo, considerando as coleções de textos que foram processadas. Além disso, embora os algoritmos baseados em medidas de centralidade tenham gerado resultados para apenas 4 coleções de textos, eles não tiveram um bom desempenho quando comparados a outros algoritmos baseados em redes.

Figura 4. Número de vitórias de cada algoritmo baseado em redes para todas as métricas de avaliação.



Na Figura 5, o número de vitórias de cada algoritmo baseado no modelo espaço-vetorial é apresentado, considerando as diferentes medidas utilizadas neste trabalho. Com base em sua observação, Analisando a figura, é possível observar que o algoritmo com o maior número de vitórias entre os algoritmos baseados no modelo espaço-vetorial é o *DBSCAN*, que também obteve o maior número de vitórias quando comparado com os outros algoritmos apresentados. Apesar disso, é possível observar que o algoritmo k-*Means*, apesar de não ter conseguido superado o algoritmo *DBSCAN* na maioria das métricas, obteve os melhores resultados para as métricas Macro-Revocação e Micro-F_1.

Figura 5. Número de vitórias de cada algoritmo baseado no modelo espaço-vetorial para todas as métricas de avaliação.



5. Conclusões

As representações baseadas em redes ganharam ênfase nos últimos anos [Tao et al. 2021, Zhang and Zhang 2020, Sun and Han 2012]. Estas representações permitem capturar padrões dificilmente capturados no modelo espaço-vetorial. Entretanto, ainda há uma lacuna na literatura quando se trata da avaliação comparativa de algoritmos baseados no modelo espaço-vetorial com algoritmos baseados em redes. Com isto, este trabalho teve como objetivo comparar os algoritmos de detecção de comunidades baseadas em redes com algoritmos de agrupamento baseados no modelo espaço-vetorial.

De acordo com os resultados, foi possível observar que o algoritmo com melhores resultados para a maioria das coleções de textos foi o algoritmo *DBSCAN* para todas as métricas de avaliação, exceto pela métrica Macro-F_1, na qual ocorreu um empate com o algoritmo k-*Means*. Entre os algoritmos baseados em redes, o algoritmo *Label Propagation* se mostrou ser o mais promissor, sendo vitorioso em todas as métricas. Além disso, os algoritmos baseados em centralidade apresentaram resultados ruins quando comparados com outros algoritmos baseados em redes para as coleções de textos avaliadas.

As técnicas e algoritmos discutidos neste trabalho podem ser aplicados a outros cenários em trabalhos futuros como: (i) detecção de novidades; (ii) organização de resultados de busca; (iii) aprendizagem de máquina baseado em uma única classe e (iv) técnica de detecção de comunidades em redes heterogêneas. Além disso, o algoritmo de agrupamento baseado em densidade, *DBSCAN*, demonstrou bons resultados para a tarefa de agrupamento de texto, o que encoraja trabalhos futuros para validar o desempenho do agrupamento de texto utilizando outras técnicas baseadas em densidade, tais como as dos algoritmos *MeanShift* [Comaniciu and Meer 2002] e *OPTICS* [Ankerst et al. 1999]. Adicionalmente, os algoritmos baseados em redes também apresentaram bons resultados para o agrupamento de textos, principalmente o algoritmo de propagação de rótulos. Portanto, ainda é necessário buscar formas de otimizar as técnicas de agrupamento para os

algoritmos baseados em redes, como a utilização de outras formas de construção de redes, diferentes medidas de distância e outras métricas de avaliação da qualidade dos grupos obtidos no agrupamento, além de explorar o agrupamento em redes heterogêneas.

Referências

Aggarwal, C. C. (2018). *Machine learning for text*. Springer.

Angelova, R. and Weikum, G. (2006). Graph-based text classification: Learn from your neighbors. In *SIGIR*, volume 2006, pages 485–492.

Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. (1999). Optics: Ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60.

Bhadane, C. and Shah, K. (2020). Clustering algorithms for spatial data mining. In *Proceedings of the 2020 3rd International Conference on Geoinformatics and Data Analysis*, ICGDA 2020, page 5–9, New York, NY, USA. Association for Computing Machinery.

Biemann, C. and Mehler, A. (2014). *Text mining: From ontology learning to automated text processing applications*. Springer.

Blanco, R. and Lioma, C. (2012). Graph-based term weighting for information retrieval. *Information retrieval*, 15(1):54–92.

Brandes, U. (2008). On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, 30:136–145.

Breve, F., Zhao, L., Quiles, M., Pedrycz, W., and Liu, J. (2012). Particle competition and cooperation in networks for semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering.*, 24:1686–1698.

Charu, C. A. and Chandan, K. R. (2013). Data clustering: algorithms and applications.

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619.

Cordasco, G. and Gargano, L. (2011). Community detection via semi-synchronous label propagation algorithms. *CoRR*, abs/1103.4550.

Freeman, L. (1977). A set of measures of centrality based on betweenness. *Sociometry*, 40:35–41.

Ganji, M., Seifi, A., Alizadeh, H., Bailey, J., and Stuckey, P. J. (2015). Generalized modularity for community detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 655–670. Springer.

Getoor, L. (2005). Link-based classification. In *Advanced methods for knowledge discovery from complex data*, pages 189–207. Springer.

Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826.

Gualdi and Rossi (2019). Aprendizado de maquina não supervisionado baseado em redes heterogeneas para agrupamento de textos. *SBC*, pages 1–13.

Hagberg, A., Swart, P., and S Chult, D. (2008). Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Jalil, A. M., Hafidi, I., Alami, L., and Ensa, K. (2016). Comparative study of clustering algorithms in text mining context. *International Journal of Interactive Multimedia and Artificial Intelligence*.

Ji, M., Sun, Y., Danilevsky, M., Han, J., and Gao, J. (2010). Graph regularized transductive classification on heterogeneous information networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 570–586. Springer.

Kanwal, S. and Asghar, S. (2021). Speech emotion recognition using clustering based ga-optimized feature set. *IEEE Access*, 9:125830–125842.

Koller, D. and Sahami, M. (1997). Hierarchically classifying documents using very few words. Technical report, Stanford InfoLab.

Massey, L. (2005). Evaluating and comparing text clustering results. In *Computational Intelligence*, pages 85–90. Citeseer.

Newman, M. (2004). Fast algorithm for detecting community structure in networks. phys. rev. e stat. nonlin. soft. matter. phys. 69(6 pt 2), 066133. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 69:066133.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*.

Raghavan, U. N., Albert, R., and Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106.

Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., Costa, L. d. F., and Rodrigues, F. A. (2019). Clustering algorithms: A comparative approach. *PloS one*, 14(1):e0210236.

Rossi, R., Marcacini, R., and Rezende, S. (2013). Benchmarking text collections for classification and clustering tasks. *Technical Report 395, Institute of Mathematics and Computer Sciences - University of Sao Paulo*.

Rossi, R. G. (2015). *Classificação automática de textos por meio de aprendizado de máquina baseado em redes*. PhD thesis, Universidade de São Paulo.

Rossi, R. G., Rezende, S. O., and de Andrade Lopes, A. (2015). Term network approach for transductive classification. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 497–515. Springer.

Sun, Y. and Han, J. (2012). Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 3(2):1–159.

Tan, P.-N., Steinbach, M., and Kumar, V. (2013). Introduction to data mining.

Tao, Y., Li, Y., and Wu, Z. (2021). Revisiting graph neural networks for node classification in heterogeneous graphs. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE.

Turner, V., Gantz, J. F., Reinsel, D., and Minton, S. (2014). The digital universe of opportunities: Rich data and the increasing value of the internet of things. *IDC Analyze the Future*, 16.

Zamir, O., Etzioni, O., Madani, O., and Karp, R. (1997). *Fast and Intuitive Clustering of Web Documents*. AAAI Press.

Zararsiz, G. (2014). High-dimensional statistical and data mining techniques. In *Encyclopedia of Business Analytics and Optimization*, pages 1117–1130. IGI Global.

Zhang, H. and Zhang, J. (2020). Text graph transformer for document classification. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. (2004). Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328.